3GPP TSG-SA WG4 Meeting #131-bis-eS4-250601r02

Online, 11 – 17 April 2025

**Source: Xiaomi**

**Title: Pseudo-CR on Video Decoder API and System Integration updates**

**Spec: 3GPP TS 26.265 v1.0.0**

**Agenda item: 9.5**

**Document for: Agreement**

**1. Introduction**

This contribution proposes updates to the Video Decoder API as well as the Systems functions.

The r01 versions implements the following changes:

* Reordering of changes by clause order of TR (despite lack of relationship between the changes in this order)
* Updating every changes against the draft integrated document [26265-102-rm.docx](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_131-bis-e/Inbox/Drafts/Video/26265-102-rm.docx)
* Adding a brief motivation in each change in addition to the general motivation

Th r02 version (based on r01\_QCOM) removes non agreeable changes based on email discussion thread 601.

**2. Reason for Change**

Addressing the case of multi-layer and multi-bitstream video representation, especially for stereo video.

The changes 1 to 4 and 5 to 6 can be addressed separately.

**3. Proposal**

It is proposed to agree the following changes to 3GPP TS 26.265 v1.0.0.

\* \* \* First Change \* \* \* \*

Motivations:

* Adding Access Unit definition which is used in 7.2.1.
* Creating the an intermediate concept between bitstream and CVS such this concept can accommodate layer in case of layer coding. The proposal is to reuse elementary stream as known from system integration, cf analysis below.
* Analysis of terminology:
  + HEVC:
    - **Elementary stream** (only 3 occurrences in the HEVC spec):
      * A sequence of one or more **bitstreams:**
        + A sequence of bits, in the form of a NAL unit stream or a byte stream, that forms the representation of coded pictures and associated data forming one or more **coded video sequences (CVSs):**

A sequence of **access units** that consists in a given order.

(general) A set of NAL units […] and contain exactly one coded picture with nuh\_layer\_id equal to 0.

(multiview) A set of NAL units [...] and contain at most one *coded picture* with any specific value of nuh\_layer\_id.

* + System (14496-1/ISOBMFF/DASH/CMAF)
    - **Elementary stream**
      * (part 1) consecutive flow of mono-media data from a single source entity to a single destination entity on the compression layer. More than one elementary stream may be connected to a single decoder (e.g., in a decoder of a scalable audio-visual object)
      * (NALU FF) sequence of one or more bitstreams of the applicable video standard
    - **access units**:
      * (part 1) smallest individually accessible portion of data within an elementary stream to which unique timing information can be attributed
      * (DASH) unit of a *media stream* (3.1.29) with an assigned Media Presentation time
      * (ISOBMFF) media data pertaining to a particular composition time in a media stream, usually carried in one sample of a media track.

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

**Access Unit:** Smallest individually accessible portion of data within an Elementary Stream to which unique timing information can be attributed.

**Bitstream:** A sequence of bits that forms the representation of any coded pictures and associated data. This sequence of bits is formed by one or more coded video sequences (CVSs) where the CVS share identical metadata.

**Coded Video Sequence:** A sequence of bits that consists of a series of coded frames and any associated metadata (required for decoder and rendering initialization) and conforms to a specific video encoding format and aligns with a certain Operation Point, as defined in this document. Such coded video sequence (CVS) has no decoding dependency on any other prior CVS and consists, in decoding order, of information specifying the characteristics or format of the encoded video data, a single intra random access coded frame followed by zero or more dependent, on the intra random access coded frame, coded frames, and a series of associated coded metadata.

Editor’s Note: Needs to be completed.

**Chroma:** a sample array or single sample representing one of the two colour difference signals related to the primary colours, represented by the symbols *Cb* and *Cr*.

**Hero Eye**: The default eye in a stereo (stereoscopic) video pair, often determined by tags set by the cameras used to capture the video.

**Luma:** a sample array or single sample representing the monochrome signal related to the primary colours (denoted with the symbol *Y*),

**Operation Point:** A collection of discrete combinations of different video representation formats, including spatial and temporal resolutions, colour mapping, transfer functions, and the encoding format.

**Receiver:** A device capable of decoding and rendering any bitstream that is conforming to a certain Operation Point.

\* \* \* 2nd Change \* \* \* \*

Not agreeable

\* \* \* 3rd Change \* \* \* \*

Not agreeable

\* \* \* 4th Change \* \* \* \*

Motivations:

* Text improvement for codecs
* Decoupling video decoder API and player API as done in CTA-5003, even if we don’t keep this reference

## 4.6 Reference API parameters

### 4.6.1 Introduction

When media is played back, the decoder and the playback pipeline need to be initialized. For this purpose, certain parameters are required. In CTA-5003 [DPC], a media playback model is described that is aligned with HTML 5.1 and the <video> element, as well as the Media Source Extensions.

### 4.6.2 Video Decoder API Parameters

Video decoders are typically accessed by API parameters. The parameters are used for the following purposes:

- to identify the capability of the device in order to check whether the signal can be played back

- to initialize the decoding and playback platform to allocate the resources for decoding and rendering

Table 4.6.2-1 provide relevant parameters for Video Decoder APIs.

Table 4.6.2-1 Video Decoder API Parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Restrictions | Status |
| media type | Specifies the media type of the component, in this case video. | required |
| codecs | Specifies through a well-defined string the codec parameters which the encoded video signal is compliant to. | required |
| video format parameters | Specifies additional video format parameters as defined in Table 4.4.2.1 to describe the signal and to initialize the encoder. | optional |

Editor’s Note: The capability of such API for decoding and playback of multilayer content, e.g. for stereoscopic content needs to be documented.

### 4.6.3 Video Encoder API Parameters

Video encoder API parameters are for further study.

### 4.6.4 Player API Parameters

Media players are typically configurable via API parameter. The main purpose of the API are:

- For video components, to create one or more display windows to display the decoded video signal

- To bind a media source, possibly remote, to the one or more created display windows.

Table 4.6.2-2 Display Window Object Parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Restrictions | Status |
| width | Specifies the width of a video player window, in pixels | required |
| height | Specifies the width of a video player window, in pixels. | required |
| video format parameters | Specifies additional video format parameters as defined in Table 4.4.2-1 to describe the signal. | optional |

Editor’s Note: The relationship between the width and height in the above table and the spatial resolution of the video signal needs be to be clarified.

\* \* \* 5th Change \* \* \* \*

Motivations:

* Most of original 601 is integrated in [26265-102-rm.docx](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_131-bis-e/Inbox/Drafts/Video/26265-102-rm.docx).
* Text improvement for RAP to change from bit position to byte position since RAP in bitstream are typically at byte position.

7.2 Functional Definitions

### 7.2.1 General

##### 7.2.1.1 Summary

This clause defines functional definitions for system integration in Table 7.2.1.1-1. The remainder of this

Table 7.2.1.1-1 Functional Definitions

|  |  |  |
| --- | --- | --- |
| Term | Summary | Details |
| Codec String | A single value identifying the codec indicated to render the content in the Bitstream as defined in IETF RFC 6381. | 7.2.1.2 |
| Decoder Configuration | a data structure storing essential parameters needed for decoding and rendering a video stream. | 7.2.1.3 |
| Random Access Point | A byte position in the Bitstream, for which in combination with the Decoder Configuration, the Bitstream can be randomly accessed, i.e. in decoding order the Bitstream carries sufficient information to access the media in the stream. | 7.2.1.4 |
| Access Unit (AU) | See Clause 3.1 |  |
| Coded access unit (CAU) | bits corresponding to an Access Unit | 7.2.1.5 |
| Random Access CAU | A CAU that starts with a random access point | 7.2.1.6 |

\* \* \* End of Changes \* \* \* \*